

Serial No. 10/044,998

Krishna Kishore Yellepeddy

Page 10 of 15

**Section III:**  
**AMENDMENT UNDER 37 CFR §1.121 to the**  
**DRAWINGS**

No amendments or changes to the Drawings are proposed.

Serial No. 10/044,998

Krishna Kishore Yellepeddy

Page 11 of 15

**Section IV:**  
**AMENDMENT UNDER 37 CFR §1.121**  
**REMARKS**

**Objections to the Specification**

In the Office Action, correction to the specification was required due to lacking a heading for the "Brief Summary of the Invention". The Brief Summary was located on page 11 following the Brief Description of the Drawings which appears on page 10, both of which have proper headings but are in reverse of the USPTO preferred order.

The present amendment relocates the Summary of the Invention to appear just before the Brief Description of the Drawings. Reconsideration of the objection is requested.

**Rejections under 35 U.S.C. §112**

In the Office Action, the examiner has rejected claims 6, 12, and 18 under 35 U.S.C. §112, first paragraph, for failing to comply with the written description requirement with respect to specifying "blocking propagation".

"Blocking propagation" is a term used in our claims as originally filed, and thus it is part of the originally filed disclosure. Our process of interrupting the normal flow (e.g. automatic forwarding or propagation of change commands) by metadirectory joiners was fully described in our paragraphs [0065] - [0067], especially in the following two sentences (emphasis added):

[0065] Figure 8 shows the high-level logical process (80) according to the invention. As previously discussed, the Joiner normally stores local copies of entries from the directories being managed by the metadirectory. When the Joiner receives an update operation (81) for an entry in a directory, it performs an "apply" operation (82) on a selected entry in the metadirectory local table, creating a temporary modified entry containing the result of the update.

[0066] This temporary modified entry is not written to the secondary storage (e.g. propagated to the other joined directories), however. The modified entry is compared (83) with the original (unmodified) entry to identify the differences between the original entry

Serial No. 10/044,998

Krishna Kishore Yellepeddy

Page 12 of 15

and the updated entry.

[0067] If there are no differences between the original entry and the updated entry (84), no updates are propagated to other directories in the metadirectory, and the temporary local entry is deleted.

Claims 6, 12, and 18 have been canceled in the present amendment. Independent claims 1, 7 and 13 have been amended to include these steps, elements and limitations, in greater detail and clarity, repeating the language of the disclosure so as to avoid discrepancy with the specification. We have employed the term "suppressing" or "suppress" as synonymous with the description of preventing propagation of the received update command, which is consistent with our disclosure and consistent with widely accepted definitions of the term "suppress", as evidenced by one dictionary (*source: [www.dictionary.com](http://www.dictionary.com)*, emphasis added):

**suppress**

1. To put an end to forcibly; subdue.
2. To curtail or prohibit the activities of.
3. To keep from being revealed, published, or circulated.
4. To deliberately exclude (unacceptable desires or thoughts) from the mind.
5. To inhibit the expression of (an impulse, for example); check: suppress a smile.
6. To reduce the incidence or severity of (a hemorrhage or cough, for example); arrest.

Serial No. 10/044,998

Krishna Kishore Yellepeddy

Page 13 of 15

**Rejections under 35 U.S.C. §103**

In the Office Action, claims 1, 2, 5, 6, 7, 8, 11, 12, 13, 14, 17, and 18 were rejected under 35 U.S.C. §103 as being unpatentable over the publication "Microsoft Metadirectory Services Concepts and Architecture (hereinafter "Microsoft") as supplied by applicant in applicant's IDS, in view of US Patent 6,615,223 to Shih (hereinafter "Shih"). Claims 3, 4, 9, 10, 15, and 16 were rejected over Microsoft in view of Shih in further view of US published patent application 2002/0038308 to Cappi (hereinafter "Cappi").

Claims 6, 12 and 18 (rejected over Microsoft in view of Shih) have been canceled, which were drawn towards our implementation of generating a "differential update" command. The details of our differential update command have been amended into independent claims 1, 7 and 13, albeit in greater detail and consistent with our specification. New Claim 19 sets forth these steps and limitations according to our preferred embodiment.

In the rationale for rejection of claims 6, 12 and 18, it was opined that our step of comparing a locally updated copy of the database entry to an original (unmodified) copy was taught by Shih at col. 14 lines 13 - 24 (our emphasis added):

Generally, the present embodiment attempts to resolve conflicts by applying the following process:

1. Attempt to detect conflict when a change is applied or upon detection of error;
2. Attempt to re-apply the change a configurable number of times or for a configurable amount of time after a waiting period;
3. If the retry limit is reached without successfully applying the change, then the change request is escalated to a different-priority queue for processing.

According to this embodiment, three change log processing queues are employed. When a change first arrives to the consumer directory, it is placed in a "new queue". An attempt is then made to apply the change. If it fails to be applied in the new queue, the change will be put

Serial No. 10/044,998

Krishna Kishore Yellepeddy

Page 14 of 15

to a "retry queue". If it fails to be applied after a specified number of attempts in the retry queue, the change will be placed to a "Human Intervention queue" and re-attempted at a much lower rate. If it succeeds to be applied from one of the above 3 queues, it will be placed to the purge queue for garbage collection.

This portion of Shih discusses detecting *conflicts* and *failed changes*, not *net differences*, which may occur due to two different directories attempting to update the same item (e.g. collision of operations), a master attempting to change an item to which it does not have permission, or commands that are 100% redundant such as adding an entry that already exists, deleting a non-existing entry, or modifying a non-existing entry, as stated by Shih (col. 13 line 13 - col. 14 line 2, our emphasis added):

Multi-master replication enables updates to multiple replication sites. Thus, a mechanism is needed to address the possibility of conflicting updates. Conflicts should be detected, for example, when the replication server attempts to apply changes from a remote directory to another directory that holds conflicting data.

Entry-level conflicts are caused when the replication server attempts to apply a change to a consumer directory that results in a conflict, such as:

adding an entry that already exists;  
deleting an entry that does not exist; or  
modifying an entry that does not exist.

Attribute-level conflicts are caused when two directories are updating the same attribute with different values, possibly at different times. One approach to address attribute-level conflicts is to examine timestamps of the changes involved in the conflict.

However, Shih is silent as to determining the net difference caused by an update command, generating a differential update command containing only the net differences, and propagating the differential update command in place of the original update command.

For these reasons, Microsoft in view of Shih, and Microsoft in view of Shih in view of Cappi fail to teach all of our claimed elements, steps, or limitations. Applicant requests

Serial No. 10/044,998

Krishna Kishore Yellepeddy

Page 15 of 15

reconsideration of the rejections and allowance of the claims.

**Conclusion**

Applicant respectfully requests entry of the amendment, and allowance of the claims for the foregoing reasons.

Respectfully,

*Robert Frantz*

Agent for Applicant(s)  
Robert H. Frantz, Reg. No. 42,553  
Tel: (405) 812-5613